

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-198563

(43)Date of publication of application : 31.07.1998

(51)Int.Cl.

G06F 9/38

(21)Application number : 10-031921

(71)Applicant : TEXAS INSTR INC <TD>

(22)Date of filing : 05.01.1998

(72)Inventor : JOHNATHAN H SHELL
GEORGE Z N KAI

(30)Priority

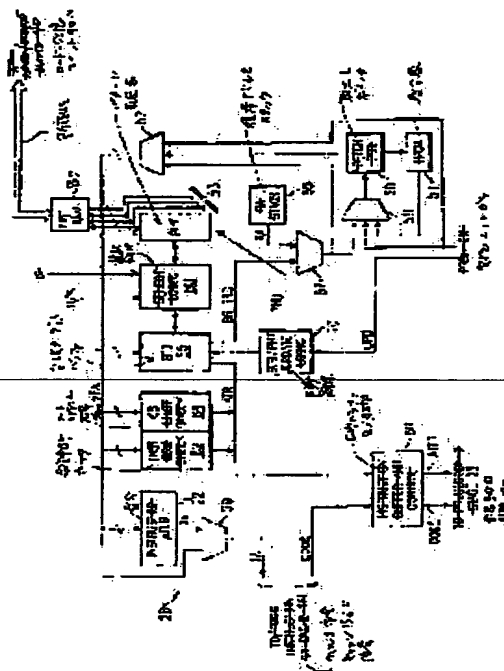
Priority number : 96 34397 Priority date : 30.12.1996 Priority country : US

(54) DYNAMICALLY LOADABLE PATTERN HISTORY CHART IN MICROPROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method for dynamically predicting the branch instruction of a microprocessor operating under a multitask environment by holding predicted information based on its own branch pattern history.

SOLUTION: A fetching unit 26 is branched with a branch target buffer 56 and plural pattern history charts 53. A selection logic 80 receives a signal indicating the kind of a program including an instruction for each branch instruction, selects one pattern history chart 53, and generates a prediction code according to a branch history field BH in the entry of the branch target buffer 56 corresponding to an instruction address by using the pattern history chart. At the time of task switching, the contents of more than one pattern history charts 53 are stored in a task state segment corresponding to an interrupted task, and an entry from the task state segment of a new task is loaded to the pattern history chart 53.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-198563

(43) 公開日 平成10年(1998) 7月31日

(51) Int.Cl.⁶

G 0 6 F 9/38

識別記号

3 3 0

F I

G 0 6 F 9/38

3 3 0 B

審査請求 未請求 請求項の数 2 書面 (全 20 頁)

(21) 出願番号 特願平10-31921
 (22) 出願日 平成10年(1998) 1月5日
 (31) 優先権主張番号 0 3 4 3 9 7
 (32) 優先日 1996年12月30日
 (33) 優先権主張国 米国 (U S)

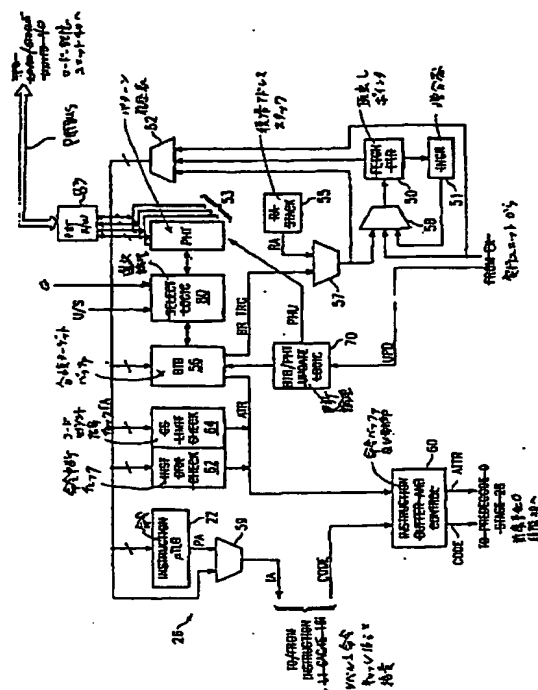
(71) 出願人 590000879
 テキサス インストルメンツ インコーポ
 レイテッド
 アメリカ合衆国テキサス州ダラス, ノース
 セントラルエクスプレスウェイ 13500
 (72) 発明者 ジョナサン エイチ. シエル
 アメリカ合衆国テキサス州プラノ, ロング
 フェロー ドライブ 4300
 (72) 発明者 ジョージ ゼット. エヌ. カイ
 アメリカ合衆国テキサス州プラノ, クラレ
 ンドン ドライブ 5720
 (74) 代理人 弁理士 浅村 皓 (外 3 名)

(54) 【発明の名称】 マイクロプロセッサ内の動的にロード可能なパターン履歴表

(57) 【要約】

【課題】 マルチタスク環境で動作するマイクロプロセッサの分岐命令を、自身の分岐パターン履歴に基づく予測情報を保持して動的に予測する方法を提供する。

【解決手段】 取出しユニット 26 は分岐ターゲットバッファ 56 と、複数のパターン履歴表 53 を有する。選択論理 80 は各分岐命令毎に、命令を含むプログラムの種類を示す信号を受けて 1 つのパターン履歴表 53 を選択し、これを用いて、命令アドレスに対応する、分岐ターゲットバッファ 56 のエントリ内の分岐履歴フィールド BH に応じて、予測コードを生成する。タスク切替えの場合は、1 つ以上のパターン履歴表 53 の内容を、中断されたタスクに対応するタスク状態セグメント 90 に記憶し、新しいタスクのタスク状態セグメントからのエントリをパターン履歴表 53 にロードする。



【特許請求の範囲】

【請求項1】 多重タスクモードで動作するマイクロプロセッサであって、

第1及び第2のタスクに従って命令を実行する少なくとも1つの実行ユニットと、

前記第1及び第2タスクのそれぞれに関連する部分を含み、また命令を記憶する部分を含む、メモリと、

メモリにアドレスして前記実行ユニットが実行する命令コードを検索する取出しユニットであって、

前記実行ユニットが実行した分岐命令の一連の結果を記憶する分岐履歴回路と、

前記分岐履歴回路に結合し、前記分岐履歴回路からの分岐履歴フィールドに対応する予測情報を与える、パターン履歴回路と、

取り出す命令のアドレスを選択する、アドレス指定回路と、を備える取出しユニットと、

前記パターン履歴回路と前記メモリに結合し、前記第1タスクから前記第2タスクへのタスク切替えに応じて前記予測情報を修正する、回路と、を備える、マイクロプロセッサ。

【請求項2】 バイプライン化マルチタスクマイクロプロセッサを操作する方法であって、

バイプライン化マイクロプロセッサの取出し段階で第1のタスクの分岐命令を検出し、

前記検出ステップに応じて、分岐履歴フィールドの少なくとも一部を検索し、

前記分岐履歴フィールドの前記検索された部分に対応する記憶された予測情報から、分岐予測を生成し、

前記第1タスクから第2のタスクへのタスク切替えに応じて、前記予測情報を修正する、バイプライン化マルチタスクマイクロプロセッサを操作する方法。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 この発明はマイクロプロセッサの分野に関し、より特定すると、バイプライン化マイクロプロセッサにおける分岐予測法に関する。

【0002】

【従来の技術】 マイクロプロセッサやその他のプログラムابل論理デバイスの分野では近年多くの改善が行われ、性能が飛躍的に改善された。その一例はバイプライン化アーキテクチャであって、多数のマイクロプロセッサ命令を種々の実行段階で同時に処理するので、前の命令が完了する前に次の命令の処理が始まる。個々の命令を処理するのに取出しから実行まで数マシンサイクルを要するにも関わらず、バイプライン方式を用いると、単一バイプラインマイクロプロセッサ内でマイクロプロセッサが命令を実行する有効速度は、マシンサイクル当たり1命令に近くなる。いわゆるスーパースカラアーキテクチャでは多数のバイプラインが並列に動作するので、理論的な性能レベルは更に高い。

【0003】 この技術で知られているように、多くの従来のコンピュータ及びマイクロプロセッサのプログラムでは分岐命令が用いられる。分岐命令はプログラムの流れを変える命令であって、分岐命令の後に実行する次の命令は、必ずしもプログラムの順序における次の命令ではない。分岐命令には、JUMP命令やサブルーチン呼出しやサブルーチン復帰などの無条件分岐命令と、前の論理や算術命令の結果に依存する条件付き分岐命令がある。

【0004】 条件付き分岐命令があるとマイクロプロセッサのバイプライン化アーキテクチャは複雑になる。それは、分岐する条件は実行するまで分からず、実行は取出してから数サイクル後になるからである。この状況では、マイクロプロセッサは分岐条件が分かるまで分岐後の命令の取出しを中止し、その結果バイプラインに空虚な段階である「バブル」（即ち、命令処理の穴）を生じるか、またはバイプラインを満たすために命令を推測により（実際には条件を推測して）取り出して、もし推測が間違ったことが分かったと現在の命令のバイプラインを「フラッシュ」するという危険を冒すか、のどちらかである。

【0005】 命令を推測により実行してバイプラインを常に満たすことの利点は、特に長いまたは多数のバイプラインアーキテクチャでは、推測による実行の成功率が所望の性能を得るのに十分である限り、バイプラインのフラッシュによる性能の低下を十分補うことである。したがって現在の多くのマイクロプロセッサは、条件付き分岐命令の行動をある程度の正確さで予測できる何らかの分岐予測法を用いている。分岐予測の種類の1つは、予測が時間や履歴に従って変わらない「静的」予測である。簡単な静的予測法は、単に全ての条件付き分岐を「行う」として予測する。より進んだ静的分岐予測法は分岐の方向に従って予測する。例えば全ての順方向の条件付き分岐を「行わない」と予測し、全ての逆方向の分岐（例えば、DOループ内のLOOP命令）を「行う」と予測する。もちろん、無条件分岐の場合は必ず静的に「行う」と予測する。

【0006】 動的分岐予測は、過去の分岐の結果を用いて次の分岐の結果を予測する、既に知られている分岐予測法である。よく知られた簡単な動的予測法は、単に最近の1つまたは2つの条件付き分岐の結果を用いて現在の分岐命令の方向を予測する。

【0007】 より正確な動的な分岐予測法は、他の命令の分岐結果ではなく自分の分岐履歴を用いて分岐命令の方向を予測する。この方法は最近のマイクロプロセッサでは、一般的に分岐ターゲットバッファを用いて行われている。従来の分岐ターゲットバッファBTBはキャッシュのエントリに似た表で、各エントリは最近出現した分岐命令の識別子（「タグ」）と、予測を行うための分岐履歴に関するコードと、分岐を行うと予測した場合に

取り出す次の命令のターゲットアドレス（その次の順次のアドレスは「行わない」予測で取り出すアドレス）を記憶する。分岐命令を取り出すと、そのアドレスをBTB内のタグと比べて、この命令が前に出現したかどうか判断する。出現した場合は、その命令についてBTB内で示された予測コードに従って次の命令を取り出す。新しく出現した分岐命令については履歴がBTB内にないので、静的に予測する。命令を実行して完了すると、BTBエントリを作り（一般に、分岐を行ったものについてだけ）または修正して（すでにBTBエントリを持つ分岐について）分岐命令の実際の結果を反映し、これを次にその命令が起こったときに用いる。

【0008】最も近く実行した分岐かまたは同じ命令の分岐履歴に基づいて分岐を予測する種々の実際の予測アルゴリズムが知られている。よく知られた簡単な分岐アルゴリズムは4状態の状態機械モデルに従い、最も近い2つの分岐事象を用いて次の分岐を行うか行わないかを予測する。4状態とは、「強く行う」と、「行う」と、「行わない」と、「強く行わない」である。「強く」の状態は、少なくとも最後の2つの分岐が全て「行う」だけか「行わない」だけの場合（実現したものに比べて、一般に、またはその特定の命令について）に対応する。「行う」状態と「行わない」状態（即ち「強く」の状態でない）は最後の2つの分岐の結果が異なる場合に対応し、次の分岐は、予測を逆の方向に変えるか、または予測の方向を保ちしかも「強く」にする。

【0009】最近の進んだ分岐予測アルゴリズムは、分岐行動を予測するのに、分岐履歴だけでなく分岐パターン情報を用いる。例えば或る分岐命令は3回通るループで、その分岐履歴は行う・行う・行わない、というパターンを繰り返す。簡単な2ビット即ち4状態の予測法を用いたのでは、その行動が完全に予測可能であってもこの命令の分岐を正しくは予測しない。よく知られた2レベル適応分岐予測法が、Yeh と pat t の「2レベル適応分岐予測（Two-Level Adaptive Branch Prediction）」、マイクロアーキテクチャに関する第24回国際シンポジウム議事録（Proceedings of the 24th International Symposium on Microarchitecture）（ACM/IEEE、1991年11月）、51-61ページ、に述べられている。この方法は、分岐履歴と共に分岐パターン情報を用いて分岐命令の結果を予測する。Yeh と pat t の方法を用いた分岐予測がBTBを用いるマイクロプロセッサアーキテクチャに適用され、英国特許出願番号第2 285 526号、1995年7月12日発行に述べられている。またこれについては、米国特許番号第5, 574, 871号を参照のこと。

【0010】上記のYeh と pat t の論文と英国特許出願番号第2 285 526号に述べられている方

法では、特有の分岐パターン毎にパターン履歴を保持し、また更新する。この方法では、パターン履歴は上述の4状態の状態機械モデルから成り、分岐パターン毎に最近の2つの分岐事象を用いて、同じ分岐パターンを持つ次に発生する分岐が「行う」か「行わない」かを予測する。（その「強く」という属性と共に）。動作を説明すると、BTB内にエントリを持つ分岐命令を検出すると、その命令の分岐履歴フィールドに含まれる分岐パターンを用いてパターン履歴表に指標を付け、そこから予測を取り出す。分岐が決定すると、その特定の命令の分岐履歴フィールドと前のパターンのパターン履歴（即ち、予測に用いた分岐パターン）を更新する。この更新されたパターン履歴を用いて、その関連する分岐パターンをBTBの分岐履歴フィールド内に持つ次の分岐命令の結果を予測する。したがって、この方法によるパターン履歴表は「グローバル」である。その意味は、命令が何かに関係なく、同じ分岐履歴パターンを持つ任意の分岐命令について分岐予測を生成する、ということである。したがって或る特定の分岐パターンについてのパターン履歴は、その分岐履歴を持つ任意の分岐命令の分岐予測の結果に基づいて定義し更新する。このようにこの基本的な2レベル法では、別の命令の分岐結果に基づいて、任意の所定の命令の分岐予測を決定する。

【0011】Yeh と Pat t の「2レベル適応分岐予測の別の方法（Alternative Implementations of Two-Level Adaptive Branch Prediction）」、コンピュータアーキテクチャに関する第19回年次国際シンポジウム議事録（Conference Proceedings of the 19th Annual International Symposium on Computer Architecture）、（ACM、1992年5月）、124-134ページ、に述べられているように、2レベル分岐予測の別の方法はこの制限に対処する。この論文の図3に示すように、この別の方法はアドレス特有のパターン履歴表を作り、BTB内の各エントリは独自のパターン履歴表を持つ。したがって、分岐命令の分岐予測は、自身の過去の履歴から生成し修正したパターン履歴に基づいて作り、同様の分岐パターンを持つ他の分岐命令の分岐結果は用いない。

【0012】

【発明が解決しようとする課題】アドレス特有のパターン履歴表を用いると同じ分岐パターンを持つ他の分岐命令から得た分岐予測に含まれる干渉はなくなるが、これを実現するコストは非常に大きい。例えば、現在のマイクロプロセッサが持つBTBは4kエントリにもなる。したがってアドレス特有のパターン履歴表に4ビットの分岐履歴の指標を用いると、それぞれが2ビット幅の16エントリを持つ4kのパターン履歴表が必要で、記憶

量は128kビットになる。したがってこの方法を実現するのに必要なチップ面積は非常に大きい。しかも、パターン履歴表の指標に分岐履歴ビットを追加して分岐予測を改善しようとする、このコストは急速に増える。例えば、分岐履歴に6ビットを用いると512kビットのパターン履歴の記憶が必要になる。それぞれの段階が一層深い多くのパイプラインをマイクロプロセッサが持つに従って、分岐の予測誤りによる損失はますます大きくなり、正確な分岐予測にかかるプレミアムは更に高くなり、アドレス特有のパターン履歴表を実現するコストは一層大きくなる。

【0013】更に別の背景として、種類が異なるマイクロプロセッサプログラムの分岐行動は、種類が同じであれば似ているが、別の種類の間では異なることが分かった。例えば、Calder と Grunwald の「ライブラリ内の分岐の予測性 (The Predictability of Branches in Libraries)」、マイクロアーキテクチャに関する第28回国際シンポジウム議事録 (ACM/IEEE、1995年11月)、24-34ページ、に述べられているように、普通使われているUNIXライブラリのサブルーチンは予測可能な分岐行動をとり、クラス即ち種類としては非ライブラリプログラムとは異なる分岐行動をとる。

【0014】更に背景として、分岐履歴とBTBのタグフィールドの一部を用いてグローバルパターン履歴表に指標を付ける方法が知られている。

【0015】また別の背景として、現在のマイクロプロセッサはマルチタスクオペレーティングシステムを支援するようになり、マイクロプロセッサは複数のタスクの間で動作を順次に切り替えて、あたかも多数のタスクを並列に実行しているように見える。一般に、例えばよく知られたx86アーキテクチャで作られたマイクロプロセッサでは、各タスクを短時間実行し、次にタスク切替えと呼ぶ事象で中断した後、別のタスクを開始または再開し、これを短時間実行し、また別のタスクに切り替えるということを続ける。このようなマルチタスク動作を行うには、各タスクのシステム文脈をタスクの中断のときに保存し、タスクの再開のときに復元しなければならない。一般にはメモリの一部を確保し、これを各タスクのシステム文脈の記憶と呼出しに用いる。x86アーキテクチャでは、タスク状態セグメント (TSS) と呼ぶシステムセグメントを各タスクに割り当てて、タスク切替えによる中断のときにその条件を記憶する。

【0016】

【課題を解決するための手段】したがってこの発明の目的は、プログラムの種類に基づくマイクロプロセッサ内の分岐予測を与えることである。

【0017】この発明の別の目的は、分岐パターン履歴表をタスク毎に保持して、ある活動的なタスク内の分岐

活動が、中断されたタスクの分岐パターン履歴に影響を与えないようにする、分岐予測を与えることである。この発明のその他の目的や利点は、図面と共にこの明細書を参照すれば当業者に明らかである。

【0018】この発明は、マルチタスクのマイクロプロセッサに1つ以上の動的に書換え可能なパターン履歴表を与えることにより、マイクロプロセッサで実現することができる。マイクロプロセッサが実行する各タスクは中断されたタスクの条件を記憶する状態セグメントをメモリ内に保持し、タスクを再開するときにその動作の条件を検索する。この発明は、タスク切替えのときに、中断されるタスクのタスク状態セグメント内のパターン履歴表の内容を記憶することにより実現される。タスクを再開するタスク切替えのときに、タスク状態セグメントに記憶した内容でパターン履歴表を書き換える。

【0019】

【発明の実施の形態】図1は、この発明の好ましい実施の形態を実現する例示のスーパースカラ・パイプライン・マイクロプロセッサ10を備える、例示のデータ処理システム300を示す。システム300とマイクロプロセッサ10のアーキテクチャは一例であって、この発明は種々のアーキテクチャのマイクロプロセッサに用いてよいものである。したがって当業者はこの明細書を参照して、他のマイクロプロセッサアーキテクチャでこの発明を容易に実現することができる。更に、この発明は単一チップマイクロプロセッサやマイクロコンピュータや多チップで実現してよく、これらの集積回路の製造は、シリコン基板、シリコン・オン・インシュレータ、ガリウム砒素、その他の製造技術により、またMOS、CMOS、パーボラ、BiCMOS、その他のデバイスを用いて実現することができる。

【0020】図1に示すように、マイクロプロセッサ10は外部バスBUSにより他のシステムデバイスに接続する。この例ではバスBUSを単一バスで示しているが、PCIローカルバスアーキテクチャを用いる従来のコンピュータで知られているように、外部バスBUSは異なる速度とプロトコルを持つ多重バスを表してよいことは言うまでもない。システム300は、次の従来のサブシステムを備える。即ち、通信ポート303 (モデムポート及びモデム、網インターフェースなどを含む)、グラフィックディスプレイ装置304 (ビデオメモリ、ビデオプロセッサ、グラフィックモニタを含む)、一般にダイナミック・ランダムアクセスメモリ (DRAM) で実現されたメモリスタック307を含む主メモリサブシステム305、入力デバイス306 (キーボード、位置決め装置、そのインターフェース回路を含む)、ディスク装置308 (ハードディスクドライブ、フロッピーディスクドライブ、CD ROMドライブを含む) などである。したがって図1のシステム300は、現在では普通になっている従来のデスクトップコンピュータやワ

ークステーションに対応すると考えてよい。当業者が認めるように、マイクロプロセッサ10の他のシステム構成もこの発明を有効に利用することができる。

【0021】マイクロプロセッサ10はバスインターフェースユニット(BIU)12を備える。BIU12は外部バスBUSに接続し、マイクロプロセッサ10とシステム300内の他の構成要素との間の通信を制御し実行する。BIU12はこの機能を実行するための制御及びクロック回路を備える。例えば、動作速度を高めるための書込みバッファや、内部マイクロプロセッサの動作の結果とバスBUSのタイミング制約を同期させるタイミング回路などである。またマイクロプロセッサ10は、システムクロックSYCLKに基づいてクロック相を生成するクロック発生及び制御回路20を備える。この例では、クロック発生及び制御回路20はバスクロックBCLKとコアクロックPCLKをシステムクロックSYCLKから発生させる。

【0022】図1から明らかなように、マイクロプロセッサ10は3レベルの内部キャッシュメモリを備える。最高レベルはレベル2キャッシュ11で、内部バスによりBIU12に接続する。この例ではレベル2キャッシュ11は統一キャッシュで、BIU12を経てバスBUSから全てのキャッシュ化可能なデータとキャッシュ化可能な命令を受け、マイクロプロセッサ10が与えるバストラフィックの多くはレベル2キャッシュ11により行われる。またマイクロプロセッサ10はキャッシュ11の周りのバストラフィックを制御して、あるバス読取り及び書込みを「キャッシュ化不可」にすることもできる。図1に示すように、レベル2キャッシュ11は2個のレベル1キャッシュ16に接続する。レベル1データキャッシュ16_dはデータ専用であり、レベル1命令キャッシュ16_iは命令専用である。マイクロキャッシュ18は、この例では完全な二重ポートレベル0データキャッシュである。主変換ルックアサイドバッファ(TLB)19は、レベル2キャッシュ11へのメモリアクセスと、BIU12を経て主メモリへのメモリアクセスを制御する。この制御は、メモリ内のアドレス変換用のページテーブルへのアクセスを整理する。TLB19はページテーブル用のキャッシュでもある。命令マイクロ変換ルックアサイドバッファ(μ TLB)22とデータマイクロ変換ルックアサイドバッファ(μ TLB)38は、レベル1命令キャッシュ16_iとレベル1データキャッシュ16_dにそれぞれアクセスするために、従来の方法で論理データアドレスを物理アドレスに変換する。

【0023】図1に示すように、マイクロプロセッサ10はスーパースカラ型であり、したがって多数の実行ユニットを備える。これらの実行ユニットは、条件付き分岐や整数や論理操作を処理する2個のALU42₀及び42₂と、浮動小数点ユニット(FPU)31と、2個のロード・記憶ユニット40₀及び40₁と、マイクロ

シーケンサ48を備える。2個のロード・記憶ユニット40はマイクロキャッシュ18への2個のポートを用いて真の並列アクセスを行い、またレジスタファイル39内のレジスタへのロード及び記憶操作を行う。この技術で知られているように、レジスタファイル39はプログラマが用いる汎用レジスタを備え、またコードセグメントレジスタCSを含む制御レジスタを備える。

【0024】これらの多数の実行ユニットは書戻しの際に、それぞれ7段階の多数のパイプラインにより制御される。パイプラインの段階は次の通り。

F 取出し: この段階は命令アドレスを生成して、命令キャッシュ即ちメモリから命令を読み取る。

PD0 前復号化段階0: この段階は最大3個の取り出されたx86型の命令の長さや開始位置を決定する。

PD1 前復号化段階1: この段階はx86命令バイトを抽出して、復号化のために固定長書式に記録する。

DC 復号化: この段階はx86命令を最小単位動作(AOps)に変換する。

SC スケジュール: この段階は該当する実行ユニット(FPU31を含む)に最大4AOpsを割り当てる。

OP オペランド: この段階はAOpsが示すレジスタオペランドを検索する。

EX 実行: この段階はAOpsと検索されたオペランドに従って実行ユニットを実行させる。

WB 書戻し: この段階は実行の結果をレジスタまたはメモリに記憶する。

【0025】図1に戻って、マイクロプロセッサ10内の種々の機能ブロックは上に述べたパイプライン段階を実行する。取出しユニット26は命令マイクロ変換ルックアサイドバッファ(μ TLB)22を用いて、後で説明する分岐予測法などにより命令ポインタから命令アドレスを生成し、レベル1命令キャッシュ16_iに与える。更に後で説明するように、取出しユニット26は線U/Sを経てコードセグメントレジスタCSから、取出しユニット26での現在の命令のプログラムの種類即ちクラスを示す信号を受ける。更に後で説明するように、取出しユニット26とロード・記憶ユニット40の片方または両方との間にパターン履歴バスPHTBUSがあり、この発明の好ましい実施の形態の取出しユニット26内の1つ以上のパターン履歴表の読取り及び書込みを行うことができる。これについては後で説明する。

【0026】命令キャッシュ16_iは取出しユニット26への命令データのストリームを作り、取出しユニット26は前復号化0段階28と前復号化1段階32に、所望のシーケンスで命令コードを与える。これらの2段階は別個のパイプライン段階として動作し、また共に動作して最大3個のx86命令を見つけてデコーダ34に与える。前復号化0段階28は3個の可変長x86命令のサイズと位置を決定し、前復号化1段階32は多バイト

命令を固定長書式に記録して復号化を容易にする。この例では、復号化ユニット34は4個の命令デコーダを備え、それぞれは前復号化1段階32から固定長のx86命令を受けて、1ないし3個の最小単位動作(AOps)を作る。AOpsは実質的にRISC命令と同じである。スケジューラ36は復号化ユニット34の出力の復号化待ち行列から最大4つのAOpsを読んで、これらのAOpsを該当する実行ユニットに割り当てる。オペランドユニット44はマルチプレクサ45を経てスケジューラ36とマイクロコードROM46から入力を受け、命令の実行に用いるレジスタオペランドを取り出す。更にこの例では、オペランドユニット44はオペランド転送を行って記憶可能のレジスタに結果を送り、またAOpsのためにロード及び記憶の種類のアドレスを生成する。

【0027】マイクロシーケンサ48とマイクロコードROM46は、ALU42とロード・記憶ユニット40が、一般に1サイクル内に実行する最後のAOpsであるマイクロコードエントリAOpsを実行するのを制御する。この例では、マイクロシーケンサ48はマイクロコードROM46に記憶されているマイクロ命令を整理して、マイクロコード化されたマイクロ命令に応じて制御を行う。マイクロコード化されたマイクロ命令の例は、複雑なまたは余り用いないx86命令や、セグメントすなわち制御レジスタを修正するx86命令や、例外や割込みの処理や、多サイクル命令(例えばREP命令や、全てのレジスタのPUSH及びPOP命令など)などである。

【0028】またマイクロプロセッサ10は回路24を備える。回路24は、JTAG走査試験の動作やいくつかの内蔵自己試験(BIST)機能を制御し、製造が完了したときやリセットなどの他の事象のときに、マイクロプロセッサ10の動作が確実であることを確認する。

【0029】次に図2を参照して、この発明の好ましい実施の形態の取出しユニット26の構造と動作を説明する。上に述べたように、取出しユニット26は復号化のために取り出す次の命令のアドレスを決定する。したがって、取出しユニット26は命令をマイクロプロセッサ10のパイプラインに読み込むシーケンスを決定し、この発明のこの実施の形態では、アドレスの推測的な実行、特に分岐予測による実行を制御する。

【0030】取出しユニット26の動作は、いくつかの方法の中からマルチプレクサ52が選択して生成した論理取出しアドレスFAに基づいて行う。取出しアドレスFAは、復号のために次の順次のアドレスを取り出すときに、単に取出しユニット26内の取出しポインタ50の内容から生成する。図2に示すように、取出しポインタ50は取出しユニット26内のレジスタであって、その出力はマルチプレクサ52の1入力と増分器51に接続する。増分器51は取出しアドレスの値を進めて次の

論理命令の値にし(スーパースカラ機の場合は、次の論理命令は必ずしも次の順次の命令ではない)、進めた取出しアドレスをマルチプレクサ58の入力に与えて、取出しポインタ50内に記憶して、またこれを次の取出しに用いる。マルチプレクサ58は、次のアクセスで用いる取出しポインタ50の更新された内容のソースを選択する。取出しアドレスFAを生成する第2の方法では、例えば取出しユニット26が予測しなかった行う分岐または予測を誤った分岐の場合に、実行ユニットの1つ(例えばマイクロシーケンサ48)からマルチプレクサ52に与える。この値もマルチプレクサ58の入力に与えて、取出しポインタ50内に記憶する。

【0031】また取出しユニット26は、プログラムシーケンスから次の取出しアドレスFAを生成する回路を備える。図2に示すように、取出しユニットは復帰アドレススタック55を備える。これはいくつかの位置を持つ後入れ先出し(LIFO)メモリで、サブルーチン呼出しとサブルーチン復帰のための復帰アドレスを記憶して、サブルーチンの推測的な実行に用いる。この発明のこの実施の形態では、取出しユニット26は分岐ターゲットバッファ(BTB)56を更に備える。これはキャッシュと同様なエントリの配置を持ち、分岐命令の現在の段階を予測するのに用いる過去の分岐の履歴を示すデータと、取出しアドレスFAとして用いる分岐命令のターゲットアドレスを記憶して、パイプラインをできるだけいつも充てん状態に保つ。この発明のこの実施の形態では、BTB56は2レベル型であり、多数のパターン履歴表(PHT)53と共に動作して、分岐パターン履歴に基づく予測コードを記憶する。これは分岐履歴情報により呼び出す。

【0032】この発明の好ましい実施の形態に関して後で詳細に述べるように、選択論理80は、特定のアドレスの分岐予測を生成するのに用いる適当な1つのパターン履歴表53を、分岐命令を含むプログラムの種類に従って選択する。図2に示すように、選択論理80は、現在の分岐命令を含むプログラムの種類に関する情報に応じて、パターン履歴表53の中から選択する。この情報は、例えばコードセグメントレジスタCSからの線U/Sにより、また対応する分岐命令のページテーブルエントリのグローバルビットからのグローバルビット線Gにより送られる。このようにして、同じ種類のプログラム(例えば、アプリケーションプログラム、共有ライブラリ、オペレーティングシステム機能)により示される分岐行動の類似性を利用することにより、分岐予測の成功率が高くなる。

【0033】この発明のこの実施の形態では、取出しユニット26は更にPHT読取り/書込み回路83を備える。回路83は、多数のPHT53のそれぞれと、またバスPHTBUSと通信する。後で詳細に説明するように、PHT読取り/書込み回路83はタスク切替えのと

きに、ロード・記憶ユニット40を経てPHT53の選択された1つの内容をメモリとの間で授受する。この発明の好ましい実施の形態では、パターン履歴表はタスク切替えのときに動的に記憶及び書換えが可能なので、特定のタスクの専用のパターン履歴を保持することができる。

【0034】パターン履歴表53内の対応する予測コードに基づいて得られる分岐予測に応じて、BTB56はバスBRTRGを通してマルチプレクサ57にターゲット命令アドレスを与える。復帰アドレススタック55はバスRAを通してマルチプレクサ57に復帰命令アドレスを与える。マルチプレクサ57の出力はマルチプレクサ52の第3入力に接続し、またマルチプレクサ58に接続して取出しカウンタ50を更新する。このようにマルチプレクサ52の3入力は次の取出しアドレスFAの3つのソースを示す。これらは物理アドレスではなく論理アドレスである。

【0035】分岐予測の結果は該当する実行ユニットから線UPDを通して更新論理70に伝えられる。後で説明するように、更新論理70はBTB56内のエントリ63の分岐履歴を更新し、また実行して評価された分岐予測が成功か失敗かに応じて、パターン履歴表53に記憶されている予測コードを更新する。

【0036】取出しアドレスFAは取出しユニット26内の種々の機能に送られ、復号化のための次の命令の取出しを制御する。例えば、取出しユニット26は命令MTLB22と通信する。命令MTLB22は、論理取出しアドレスFAと一致する物理アドレスPAが前に変換された位置を指す場合は、これを戻す。または論理取出しアドレスFAは、取出しユニット26の外部にある主変換ユニット（図示せず）により物理アドレスに変換される。どちらにしても、取出しユニット26は命令線アドレスIAをレベル1命令キャッシュ161に与えて、ここから一連の命令コードを検索する。もちろん、レベル1命令キャッシュ161でキャッシュミスが起こった場合は、物理アドレスを統一レベル2キャッシュ11に与え、またキャッシュミスがこのレベルで起こった場合は、主メモリに与える。命令線アドレスIAに応じて、レベル1命令キャッシュ161は一連の命令コードシーケンスCODEを取出しユニット26内の命令バッファ及び制御60に与え、最終的に前復号化0段階28に与える。この場合は各命令線アドレスIAを用いて16バイトのブロックにアドレスするので、命令バッファ及び制御60の容量は少なくとも16バイトである。

【0037】また取出しユニット26は、中断命令であると識別された命令を更に取り出さないようにする命令中断チェック回路62などの他の従来の機能も備える。また取出しユニット26は、論理取出しアドレスFAが現在のコードセグメントの境界の外のアドレスを指すかどうか判断する、コードセグメント限界チェック回路6

4を備える。

【0038】論理取出しアドレスFAはBTB56の入力に入る。BTB56は、取出しアドレスFAが最近取り出した分岐命令を指しているか、また推測的な（speculative）実行に用いる分岐履歴をBTB56内に記憶しているか、を判断する。この技術で知られているように、推測的な実行は、図1のスーパーカラ・マイクロプロセッサ10などのように深くパイプライン化されたマイクロプロセッサの性能を高める重要な方法である。それは、予測を誤って分岐すると（パイプラインは機能を停止して条件付き分岐の結果を待つ）、実行機会が失われるので大きな損失を生じるからである。BTB56はキャッシュに似た構成に配置されたメモリで、例えば512エントリの、4通りのセットアソシエティブ・キャッシュバッファである。もちろんBTB56は、直接マッピングからフル連想型まで、任意の方法で構成することができる。次に図3を参照して、選択論理80及び多数のパターン履歴表53の例と組み合わせて、BTB56の構造を説明する。

【0039】上に述べたように、この例のBTB56は、多数のエントリ63を有する4通りのセットアソシエティブ・キャッシュメモリである。簡単のために、図3では1通りだけを示している。BTB56はセレクトタ61を備える。セレクトタ61は線FAを通して取出しアドレスを受け、取出しアドレスが指すエントリ63を選択する。セレクトタ61は従来の任意の方法で作り、例えばデコーダやタグ比較器や簡単なマルチプレクサにより、取出しアドレスからBTB56内のエントリ63を選択する。BTB56内の各エントリ63は特定の分岐命令の論理取出しアドレスFAにより各エントリ63を識別するのに用いるタグフィールドTAGを持ち、セレクトタ61はこれと入力する論理取出しアドレスFAの一部とを比較する。この技術で知られているように、タグフィールドTAGは対応する分岐命令の論理取出しアドレスFAの選択されたビットを直接記憶し、またはこれらの選択された論理アドレスビットの論理的組合わせに対応する。一般にタグフィールドTAGは線アドレスと、取出し線内の命令のバイトオフセットを示すオフセットを含む。またBTB56内の各エントリ63は、分岐命令ターゲットアドレスの論理アドレスを含むターゲットフィールドTARGETを有する。上述のように、「行う」と予測された分岐命令と一致する、エントリ63のTARGET部内のターゲットアドレスは、BTB入出力論理69からバスBRTRGを通してマルチプレクサ57に送られる。分岐が「行わない」である場合は、マルチプレクサ52は、単に次の順次の論理アドレス（即ち、取出しポインタ50の内容）を次の論理取出しアドレスFAとして選択する。

【0040】またこの発明のこの実施の形態の各エントリ63は、タグフィールドTAGに対応する分岐命令の

分岐履歴を記憶するmビットの分岐履歴フィールドBHを含む。分岐履歴フィールドBHに記憶される分岐履歴は、命令の実行が終わったときに決定される関連する分岐命令の実際の分岐履歴と、まだ実行が終わっていない分岐命令の事例の予測結果から成る推測的な分岐履歴を含む。更に、同時係属出願の米国仮出願番号第60/020,844号、1996年6月28日出願（ここに参照することにより挿入する）に述べられているように、BTB56内の各エントリ63は分岐履歴フィールドBH内の推測的な分岐履歴ビットの数を示すカウンタも備え、予測誤りから回復するのに用いる。またBTB56内の各エントリ63は標識TYPEを含む。これはその関連する命令の分岐命令の種類（即ち、条件付き分岐、CALL、JUMP、RETURN）を記述するもので、分岐を予測するのに用いる。CALL、JUMP、RETURNなどの無条件分岐は常に「行う」と予測する。LRUビットや有効ビットやその他の制御ビット（図示せず）などの追加のビットもBTB56の各エントリ63内に与えられる。

【0041】図2に関して上に述べたように、多数のパターン履歴表（PHT）53を用いて、選択されたBTBのエントリ63の分岐履歴フィールドBHの最新のkビットに基づいて条件付き分岐の行動を予測する。この発明のこの実施の形態では、各PHT53は特定の種類のプログラムに関連し、分岐履歴フィールドBHはPHT53の任意の1つにアクセスすることができる。しかし予測コードは、命令を取り出した種類のプログラムに該当する1つのPHT53だけから選択する。図2ではPHT53とBTB56は物理的に別の回路で実現しているが、もちろん必要に応じてPHT53をBTB56に含めてよい。図3で明らかなように、この例では4個のPHT53から530をBTB56と組み合わせて実現している。

【0042】各PHT53は簡単なルックアップメモリであって、それぞれはBTB56の選択されたエントリ63からkビットの分岐履歴を受けるセクタ67を備え、これに対応する2k個の予測エントリPRDの中の1つを選択する。セクタ67はデコーダまたはマルチプレクサで実現してこの機能を実行する。図3に示すように、PHT53から530はそれぞれ選択されたエントリ63からkビットの分岐履歴を受け、与えられたkビットの分岐履歴に対応するエントリPRDの内容に対応する一組の線PRE₃からPRE₀を通してパターン履歴コードを出す。選択された1つのPHT53に指標を付ける際に、例えばいくつかのアドレスヒットや制御情報などの他の情報を、分岐履歴フィールドBHのこれらのkビットと組み合わせる。この発明のこの実施の形態では、各組のPREは2線を用いて、4状態の分岐予測モデル（即ち、強く行う、行う、行わない、強く行わない）の下に、従来の方法で2ビットのパターン

履歴コードを送る。

【0043】この発明のこの実施の形態では、選択論理80はBTB56に与えられる選択された1つのPHT53の出力を送る回路を備える。もちろん選択論理80は多数のPHT53の中の適当な1つに選択的にアドレスするようにして実現してよい。この例では、PHT53乃至530からパターン履歴線PRE₃乃至PRE₀をそれぞれマルチプレクサ68の入力に与える。マルチプレクサ68はパターン履歴線PRE₃乃至PRE₀の組の1つを選択して、線TNTを通してBTB入出力論理69に与え、これから適当な分岐予測を行う。この発明のこの実施の形態では、マルチプレクサ68は、現在の分岐命令に対応するページテーブルエントリPTE（後で詳細に説明する）内のグローバルビットGの状態に応じて、また線U/S上の信号に従って制御される。この例では、線U/S上の信号は、マイクロプロセッサ10のコードセグメントCSレジスタに含まれる現在の特権レベル（CPL）の状態に対応する。この例では、マイクロプロセッサ10はx86アーキテクチャに従って作る。後の説明から明らかなように、マルチプレクサ68は分岐予測を行うのに用いる線PRE₃乃至PRE₀の適当な組を、分岐命令を出すプログラムの種類即ちクラスに従って選択する。これについては以下に説明する。

【0044】x86アーキテクチャでは、マイクロプロセッサ10が実行するプログラムはカーネル（最高の特権）からアプリケーション（最低の特権）まで、異なる特権レベルに従って分類される。したがって、個々の命令は、種々の特権レベルに従ってアクセスが保護されるメモリ部分に常駐する。これにより、マルチタスク環境で動作する多数のアプリケーションプログラムはプログラムやサブルーチンを共有することができる。x86アーキテクチャのメモリページング保護機構では、メモリのこれらの部分をユーザ及びスーパーバイザレベルと呼ぶ。ユーザ保護レベル（CPL=3）はアプリケーションプログラムが記憶されているメモリ位置に割り当て、スーパーバイザ保護レベル（CPL=0から2）はオペレーティングシステムや拡張やドライバやカーネルが常駐するメモリ位置に割り当てる。したがってこの例では、線U/S上の信号は、コードセグメントCS内のCPLの値に基づいて、現在の分岐命令を含むプログラムの特権レベルを示す。

【0045】もちろん現在の分岐命令が関連するプログラムの種類は、他の方法で、例えばx86アーキテクチャ内のCPLに対応する多数の信号線により、または他のアーキテクチャに従うマイクロプロセッサ内の他の種類の信号により、示してよい。どちらにしても、マルチプレクサ68は現在のプログラムの種類に対応する少なくとも1つの信号に従って制御され、分岐行動は異なる種類のプログラムの分岐命令によって異なってよい。こ

の発明の好ましい実施の形態では、BTB56とパターン履歴表53は、アプリケーションプログラム内の分岐命令（ユーザレベルのメモリに常駐している命令）の分岐行動の類似性や、オペレーティングシステム内の分岐命令（スーパーバイザレベルのメモリに常駐している命令）の分岐行動の類似性や、両方のレベルの共有ルーチン内の命令の分岐行動の類似性や、これらの異なる種類のプログラム内の分岐命令間の分岐行動の非類似性などを利用する。これを実現するために、この発明の好ましい実施の形態では、少なくとも1つのPHT53をユーザレベルの分岐命令に関して用いるように、また少なくとも1つの他のPHT53をスーパーバイザレベルの分岐命令に関して用いるように割り当てる。この実施の形態では、2つのPHT53をユーザレベルの分岐に割り当て、他の2つのPHTをスーパーバイザレベルの分岐命令に割り当てる。この発明のこの実施の形態では、線U/S上の信号をマルチプレクサ68の制御入力1つに与えてこの制御を行う。

【0046】上に述べたように、線U/Sの状態と他の制御フラグ及びビットを組み合わせ、これを用いて適当なPHT53を選択してよい。マイクロプロセッサの分野でよく知られているように、また上に述べたように、ページテーブルエントリを用いて論理アドレスから物理アドレスへのアドレス変換を行う。上に述べたようにマイクロプロセッサ10では、TLB19はページテーブルエントリPTEのキャッシュとして動作する。それぞれは、現在のアドレスが写像するページフレームアドレスを含むだけでなく、アドレスが指すメモリのページフレームに関するこの技術で知られているいくつかの制御情報も含む。ペンティアム（R）プロファミリー・デベロッパーズマニュアル（Pro Family Developer's Manual）、第3巻、オペレー

ティングシステム・ライターズガイド（Operating System Writer's Guide）（インテル、1996年）の3-21ページから3-26ページ（参照することによりここに挿入する）に述べられているように、PENTIUM PROマイクロプロセッサのアーキテクチャに従うページテーブルエントリはグローバル（ページ）ビットGを含む。このビットが、セットされているときは、変換ルックアサイドバッファ内のページエントリはタスク切替えのときにクリアされないことを示す。これにより、いくつかのタスクからのアクセスが可能な、メモリの共通ページを割り当てることができる。

【0047】例えば、C++言語で書かれたプログラム用のライブラリルーチンは、多数のC++タスクからアクセスできるようにグローバルメモリページ内に記憶する。Calder と Grunwaldの論文に関して上に述べたように、ライブラリルーチンは他の種類のプログラムとは異なる分岐行動を持つことが分かった。したがって、適当なPHT53の選択にグローバルビットを用いれば、ページテーブルエントリにグローバルビットを持つマイクロプロセッサに有用である。この発明のこの実施の形態では、ページテーブルエントリPTE内のグローバルビットGの状態（上に述べたように、タスク切替えのときにそのページエントリがTLB19からクリアされるかどうかを示す）は、上に述べたコードセグメントCSから線U/Sで送られて来る信号と共に、マルチプレクサ68の制御入力に入る。

【0048】この発明の好ましい実施の形態の、マルチプレクサ68によるPHT53から530の選択の例を次の真理値表に示す。

【表1】

<u>ユーザ/スーパーバイザ</u> <u>User/Supervisor</u>		<u>グローバルビットGの状態</u> <u>Global bit G state</u>	<u>PHT selected</u> 選択したPHT
Supervisor	スーパーバイザ	0	53 ₀
Supervisor	スーパーバイザ	1	53 ₁
User	ユーザ	0	53 ₂
User	ユーザ	1	53 ₃

【0049】この発明の好ましい実施の形態では、マルチプレクサ68の制御は、線U/S上のユーザ/スーパーバイザ状態と、グローバルビットGの状態に応じて行うが、適当なPHT53を選択するのに他の制御信号または情報をこれらの代わりにまたは追加して用いてよい。例えば、線FA上の取出しアドレスの一部と書き込み可能な範囲レジスタの内容とを比較して、取出しアドレスが、範囲レジスタが示す範囲内か範囲外かを決定し、適当なPHT53を選択する際のプログラムの種類の識

別子としてこれを用いてよい。または、ページテーブルエントリPTE内のこれまで割り当てられていない他のビットを用いて、適当なPHT53の選択のプログラム制御を行うことができる。更にまた、セグメント記述子DESC内の1つ以上のビットの状態に従ってこの選択を行ってもよい。セグメント記述子DESCは、x86アーキテクチャのマイクロプロセッサの保護モード動作中にセグメントセレクトが指標付けした、グローバルまたはローカル記述子テーブル内のエントリである。これ

らのビットは、現在まだ定義されていないビットか、またはセグメント記述子の拡張から得られたPHT53の選択コードを与えるビットである。更にまた、上述の制御信号と取出しアドレス自身の選択されたサブセットとを組み合わせ、分岐予測に用いる適当なPHT53を選択してよい。多数のPHT53から選択するこれらの方法は、出願人の仮出願番号第 /号、1996年12月10日出願、「マイクロプロセッサ内で分岐予測に用いる多数のグローバルパターン履歴表(Multiple Global Pattern History Tables for Branch Prediction in a Microprocessor)」(代理人書類番号第T1-23791P)に詳細に述べられており、これを参照することによりここに挿入する。

【0050】図3に戻って、前に述べたように、線TNT上のマルチプレクサ68の出力はBTB入出力論理69に与えられる。線TNT上の予測コードが「予測された行」分岐を示す場合は、BTB入出力論理69はBTB内の現在のエントリ63のTARGET部に対応する有効な分岐ターゲットアドレスを与える。またBTB入出力論理69は命令バッファ及び制御60への線ATRに、現在の命令の対応する分岐予測を出す。更にBTB入出力論理69は、実行ユニットからの線NEWNにより、新しく出現した分岐命令の該当するタグ、ターゲット、オフセット、種類、履歴情報を受け、選択されたエントリ63にこの情報を従来の方法で書き込む。この技術で知られているように、LRUなどの制御ビットを用いて、新しい命令の情報を書き込むエントリ63を選択する。

【0051】またBTB56は更新論理70を備える。更新論理70は、前に予測された分岐命令の結果を示す実行ユニット(例えば、ALU42)からの信号をバスUPDに受ける。更新論理70は従来の回路で、関連する分岐の予測が正しかったか誤りだったかに従って、BTB56内のエントリ63の内容を更新する。更に、PHT53が適応性を持つことを考慮して、更新論理70はPHT53への線PHUを駆動して、従来の方法により、実行された分岐命令の予測の結果に従って予測コードエントリPRDの内容を更新する。しかしこの発明の好ましい実施の形態では、更新論理70が生成した線PHU上の信号は、多数のPHT53の中から、終わった分岐について更新すべき適当なものを選択する。パターン履歴表の適応更新はこの技術で知られており、例えば前に参照したYeh と patt の論文に述べられている。

【0052】この発明の好ましい実施の形態のBTB56と多数のグローバルパターン履歴表53の動作を、図3に関して以下に説明する。もちろん、BTB56は線FAのアドレスで取り出された非分岐命令では動作しな

い。最近出現しなかった(そしてBTB56内に割り当てられた有効なエントリ63をこの時点に持たない)分岐命令については、セクタ61はエントリ63のどのTAGフィールドにも一致するタグを見出さないで、誤り信号、即ち「失敗」信号を命令バッファ及び制御60への線ATRに返す。この場合は、有効な分岐ターゲットアドレスはマルチプレクサ57へのバスBR TRGに与えられず、マルチプレクサ52は次の論理取出しアドレスFAのために別のソース(一般に取出しポイント50)を選択する。この分岐命令の実行段階が終わると、BTB56は線NEWNから得た情報を用いて、BTB入出力論理69を経て従来の方法で更新され、有効なエントリ63がこの分岐命令に割り当てられる。

【0053】前に出現した、したがって対応するエントリ63をBTB56内に持つ(即ち、取出しアドレスFAの一部がエントリ63のTAGフィールドと一致し、対応するエントリ63のTYPE部から分かる)無条件分岐命令では、BTB56は命令バッファ及び制御60への線ATRに「行」予測を与え、またマルチプレクサ57へのバスBR TRGにこのエントリ63のTARGETフィールドからターゲットアドレスを与え、マルチプレクサ52はこれを次の命令アドレスのソースとして、従来の方法で用いる。これも無条件分岐命令であるサブルーチンRETURN命令の場合は、マルチプレクサ57は復帰アドレススタック55からの線RA上の適当な復帰アドレスを選択し、マルチプレクサ52に次の命令アドレスのソースとして、従来の方法で与える。

【0054】BTB56のセクタ61が、線FAで送られる現在の取出しアドレスが有効なエントリ63を持つ条件付き分岐命令に対応すると決定した場合は、BTB56は有効なエントリ63のkビットの分岐履歴フィールドBHを各PHT533から530に送る。これらのkビットはその分岐命令の最近のk個の予測に対応し、実際の分岐結果だけを含むこともあり、またはまだ評価が済んでいない推測的な分岐予測も含むこともある。選択されたエントリ63のこれらのkビットの分岐履歴フィールドBHを、一般に現在の分岐命令の現在の分岐パターンと呼ぶ。この発明の好ましい実施の形態では、各PHT533から530内のセクタ67はこれらのkビットを復号して、現在の分岐パターンと一致する予測コードエントリPRDを選択し、選択された予測コードエントリPRDの内容を、関連する出力線PRE3からPRE0によりマルチプレクサ68に送る。各予測コードエントリPRDは好ましくは2ビットコードを含み、行、行わない、強く行、強く行わない、の4つの予測状態の中の1つを示す。

【0055】一方、分岐命令を含むメモリの対応する部分に関するページテーブルエントリPTEj内のグローバルビットGの状態と、線U/Sの状態は、マルチプレクサ68を制御して一組の出力線PRE3からPRE0

を選択し、線TNTを経てBTB56のBTB入出力論理69に与える。上に述べたように、線TNTは好ましくは、行う、行わない、強く行う、強く行わない、の予測状態の1つを示す2ビットコードを送る。次にBTB入出力論理69は線TNT上のコードに基づいて予測を得、この予測（「行う」または「行わない」）を線ATRで命令バッファ及び制御60に送る。予測が「行う」の場合は、対応するエントリ63のTARGETフィールドをバスBTRGに与え、マルチプレクサ57と52はこれを次の論理取出しアドレスFAとして選択する。予測が「行わない」の場合は、バスBTRGに有効なターゲットアドレスを与えず、マルチプレクサ52は取出しポイント50の増分された出力を次に取り出す命令のアドレスとして選択する。予測を生成した後で、かつBTB56がエントリ63の分岐履歴フィールドBH内に推測的な分岐履歴を記憶している場合は、更新論理70は現在の命令に対応するエントリ63内の分岐履歴フィールドBHを更新する。命令に関する識別情報と、予測の生成に用いるBTB56内のエントリ及びPHT53に関する識別情報と共に、現在の分岐命令に関する予測情報も、パイプラインにより命令と共に送られる。または現在の分岐命令の小さな識別子をパイプラインと共に送ってよい。この場合はこの識別子は、BTB56と該当するPHT53を更新するのに用いる取出しユニット26のまたはその近くのローカル記憶内の位置を指す。

【0056】分岐命令が終わると、該当する実行ユニットは分岐の実際の結果を線UPDを通して更新論理70に送る。更新論理70は終わった命令に対応するBTB56のエントリ63内の分岐履歴フィールドBHに向けて適当な信号を生成して、対応する予測が正しいか正しくないか検査する。更に更新論理70は分岐の実際の結果に従って、線PHUにより、該当するPHT53の中の該当する予測コードエントリPRDを更新する（必然的に線PHUには正しいPHT53と該当するエントリPRDを選択するのに必要な信号が送られる）。

【0057】図3に示すようにこの発明の好ましい実施の形態では、PHT読取り／書き込み回路83の制御の下に、PHT53はバスPHTBUSを経て読取り可能かつ書き込み可能である。この発明のこの実施の形態では、PHT読取り／書き込み回路83は双方向マルチプレクサ82とPHTアクセス制御回路84を備える。マルチプレクサ82の片側はバスPHTBUSに接続し、その反対側は個々のバスによりPHT53に接続する。この例では、32ビットバスによりマルチプレクサ82とPHT53を接続し、1つのPHT53の全内容を1動作で送ることができる。またはより小さなバス（2ビットバスPREを含む）を用いて、複数の読取りまたは書き込みサイクルにより、選択されたPHT53の内容をマルチプレクサ82に送ってよい。この場合は、好ましくはマ

ルチプレクサ82と共に別のレジスタを設けて、選択されたPHT53の全内容をバスPHTBUSのデータ線に乗せる。

【0058】この例では、マルチプレクサ82は32ビットバスにより各PHT53₁からPHT53₃に接続する。上に述べたように、PHT53₀はスーパーバイザレベルのプログラムに関連するがグローバルではない。したがってこの発明のこの実施の形態では、PHT53₀は好ましくは書き込み可能でない。それは、この性質のプログラムの分岐パターン履歴に基づく予測コードは、BTB56内に保持するのが好ましいからである。もちろん別の方法として、図3に点線のバス線で示すように、PHT53₀をマルチプレクサ82に同様に接続してよい。各PHT53₁からPHT53₃は書き込み可能であって、図3に示すようにマルチプレクサ82に接続する。PHT53₂と53₃は、関連するユーザ特権レベルを与えられた種々のタスクで動的に書き換えるのに特に適している。更に、PHT53₁は、関連するグローバルスーパーバイザレベルのプログラム（一般にライブラリ）が共通の分岐行動を持つという特殊な場合だけ書き換えてよい。特に、PHT53₁の内容を保存したり再ロードしたりするという特殊な場合は、ライブラリルーチンの種類がタスクによって変わる場合に対応する（例えば、C++タスクとCOBOLタスクの間で切り替える）。

【0059】PHTアクセス制御回路84はマルチプレクサ82を制御して、バスPHTBUSのデータ線と読取りまたは書き込む選択されたPHT53を接続し、また関連する読取り／書き込み線R/Wを経て選択されたTPHT53を制御して、読取りまたは書き込みを行わせる。PHT53の選択と、読取りと書き込みのどちらを行うかの選択は、バスPHTBUSの制御線によりPHTアクセス制御回路84に伝えられる。この発明の好ましい実施の形態では、PHTアクセス制御回路84は選択された読取り／書き込み線R/Wに該当する信号を出し、選択されたPHT53は読取りか書き込みかによって、その全内容をマルチプレクサ82に送り、またはマルチプレクサ82から新しい内容を受ける。またPHTアクセス制御回路84はオペランドユニット44に適当なハンドシェイク信号を送り、バスPHTBU上のデータの送信を制御する。

【0060】この発明の好ましい実施の形態では、1つ以上のPHT53の内容の読取りと書き込みは、好ましくはタスク切替えのときに行う。マルチタスクのマイクロプロセッサの技術で基本的なことであるが、タスク切替えは、現在の活動的なタスクを他のタスクにより中断する事象である。中断されたタスクはタスク切替えのときにその全ての条件を、一般にこの技術でタスク制御構造と呼ぶ（またはタスク制御ブロックとかタスク状態ブロックとも呼ぶ）メモリの一部に保存する。特定すると、

上述のようにマイクロプロセッサ 10 を実現する x86 アーキテクチャでは、タスク制御構造は、中断されたタスクの条件を記憶するタスク状態セグメント (TSS) を含む。タスク状態セグメントの特定の位置は変動してよく、一般に主メモリ 305 内に常駐し、実行中の便宜のためにその写しをレベル 2 キャッシュ 11 やその他のキャッシュに記憶する。後でタスク切替えを行って前に中断されたタスクを再開するときなどに、そのタスクの TSS の内容を検索してマイクロプロセッサの適当なレジスタや記憶に読み込み、タスクを再開する。この技術でよく知られているように、任意の時点で活動的なタスクは 1 つだけであるが、このようなマルチタスク動作を行うとマルチ処理のように見える。

【0061】この発明の好ましい実施の形態では、タスク切替えのときに、1 つ以上の PHT 53 の内容を TSS の一部に記憶し、タスクを再び活動的にするタスク切替えのときにそこから検索して、対応する PHT 53 に再び読み込む。このようにして、この発明の好ましい実施の形態のマイクロプロセッサ 10 は、分岐パターン履歴に基づく分岐予測情報をタスク毎に保持するので、より正確に分岐予測を行うことができる。

【0062】図 4 を参照して、マイクロプロセッサ 10 が実行する特定のタスクに関連する、この発明の好ましい実施の形態の例示の TSS 90 のメモリマップを詳細に説明する。もちろんマイクロプロセッサ 10 が実行する各タスクは自身の TSS 90 をメモリ内に持ち、各 TSS 90 は、図 4 に示すように、またこの発明の好ましい実施の形態に従って説明するように配列される。TSS 90 内の各語は TSS ベースアドレスから或るオフセットにある。この例では、TSS 90 はメモリの中の 30 語を占め、オフセットは TSS 90 のベースアドレスから最大 29 語である。TSS 90 の内容の多くは、特に 25 語以下のオフセットにある TSS 90 の内容は、インテル社製の PENTIUM マイクロプロセッサの機能性を持つマイクロプロセッサでは従来からあるものである。図 4 に示すように、TSS 90 は完全なオフセット及びセグメントレジスタの内容と、異なる特権レベル (CPL0 から CPL2) のスタックの ESP ポインタ及びセグメント SS と、タスクのページディレクトリのベースアドレスを記憶する CR3 レジスタの保存された内容を含む。また TSS 90 は、I/O マップベースエントリを含む。これは、保護モードにおいて I/O アドレス空間の保護に用いる I/O マップのアドレスである。前の TSS に逆に連結するためのエントリはセグメント記述子を含む。これは、タスクが互いに入れ子になっているときに前の中断されたタスクの TSS を参照するものである。T ビットはデバッグトラップビットで、これがセットされると、タスク切替えの時にデバッグ例外になる。TSS 90 の中の N/U と示されているフィールドは用いない。

【0063】上に述べた従来の TSS エントリの他に、TSS 90 はこの発明の好ましい実施の形態の、分岐パターン履歴に基づく予測情報の記憶と検索に用いる別のエントリを含む。これらの別のエントリは TSS ベースアドレスから 26 語オフセットから始まる。TSS ベースアドレスから 26 語オフセットの下位部分は、動的ロード制御ビット DLB を有するエントリ 92 を含む。TSS のエントリ 92 内の DLB のビット数は書き込み可能な PHT 53 の数に対応し、DLB の各ビットは 1 つの PHT 53 に関連する。PHT 53₁ から 53₃ だけが書き込み可能 (PHT 53₀ は書き込み不可) である図 3 の例では、エントリ 92 内に 3 ビットの DLB が与えられる。後で詳細に説明するように、DLB の各ビットは、TSS 90 に関連するタスクにタスク切替えを行うときにその関連する PHT 53 に専用のパターン履歴データをロードするかどうかを示す。一般に DLB のビットは、必要に応じてタスク自身による、またはオペレーティングシステムによる命令制御の下にセットされる。

【0064】また TSS 90 は、この例で TSS 90 内の 27 語から 29 語オフセットにあるエントリ 91₁ から 91₃ を含む。これらは書換え可能な PHT 53₃ から 53₁ にそれぞれ対応する。この例でエントリ 91 は 32 ビットのサイズを持ち、それぞれ関連する PHT 53 の内容を記憶する。その内容は、TSS 90 に関連するタスクを中断した最近のタスク切替えのときの条件である。上に述べたように、この場合は分岐履歴の 4 ビットを PHT 53 に指標付けし、また各 PHT 53 はそれぞれ各 2 ビットの 16 エントリを含んでいるので、1 つの PHT 53 の全内容を記憶するには 32 ビットの記憶で十分である。

【0065】26 語オフセットにある語の高位部分のエントリ 94 は、PHT 53 の内容を含む TSS 90 の部分の、ベースアドレスに対応するフィールドを含む。したがって、エントリ 91 のベースアドレスは TSS 90 のベースアドレスとエントリ 94 の内容の和のところである。図 4 に示す例では、エントリ 94 の内容は 27 語オフセットに対応する。または、エントリ 91 の位置は TSS 90 の中の別の場所でもよい。この場合は、エントリ 94 の内容は、TSS 90 のベースアドレスに対するエントリ 91 の位置を示す。

【0066】次に図 5 を参照して、この発明の好ましい実施の形態のタスク切替えルーチンの一部の動作を説明する。図 5 の動作は、タスク切替えに必要な適当なシステムレベルの動作、例えば中断されたタスクに関する TSS 内の機械条件の情報の記憶や、新たに活動化するタスクに必要な TSS からの機械条件の再ロードなど、を行う一連のプログラム命令内に含まれる。したがって、図 5 の動作はマイクロプロセッサ 10 内の制御及び実行回路により行われる。これは従来のオペレーティングシステムの命令シーケンスで一般的である。以下の説明の

便宜上、中断されたタスクをTSS90_iに関連したタスク_iと呼び、新たに活動化するタスクをTSS90_jに関連したタスク_jと呼ぶ。

【0067】図5に示すタスク切替えプロセスの部分は決定95から始まる。ここで、中断されるタスク_iのTSS90_i内のビットDLBの状態を調べる。上に述べたように、TSS90_i内のLDBのビットがセットされているときは、DLBのセットされたビットで示されるPHT53については、タスク_iは活動的なときの分岐予測に自身の分岐パターン履歴に基づく予測情報を用いることを示す。したがって、専用の予測情報を他のタスクにより修正されるのを防ぐために、示されたPHT53の現在の条件をTSS90_iに記憶する。したがって、TSS90_i内のDLBの任意のビットがセットされている（即ち、決定95がYESである）場合は、プロセス96を行う。図3を参照すると、プロセス96でPHTアクセス制御84はマルチプレクサ82を制御して、TSS90_i内のDLBのセットされているビットにより示されたPHT53を順次にバスPHTBUSに接続し、また選択されたPHT53に関連するセクタ67を制御して、その関連するPHT53の内容を順次に読み取る。これらの内容を順次にバスPHTBUSに乗せ、メモリ内の、中断されたタスク_iに関連するTSS90_iの対応するエントリ91に記憶する（ロード・記憶ユニット40の1つにより）。上に述べたように、この情報を記憶するTSS90_i内のアドレスは、TSSベースアドレスと、エントリ94のPHTエリアベースの和から決定する。次に流れは決定97に進む。TSS90_i内のDLBのどのビットもセットされていない（即ち、決定95がNOである）場合も、流れは決定97に進む。

【0068】決定97では、開始（または再開）するタスク_jのTSS90_j内のDLBのビットの状態を決定する。TSS90_j内のDLBのどのビットもセットされていない（決定97がNOである）という条件は、タスク_jがPHT53の内容を現在の条件で用いることを示す。したがって、どのPHT53も操作せずに流れはタスク切替えルーチンに進む。TSS90_j内のDLBの1つ以上のビットがセットされている（決定97がYESである）場合は、タスク_jは自身の分岐パターン履歴に基づく予測情報を持ち、これをその分岐命令の分岐予測に用いる。この場合は、流れはプロセス98に進む。

【0069】プロセス98で、ロード・記憶ユニット40の1つがタスク_jのTSS90_jの対応するエントリ91の内容をバスPHTBUSに乗せると同時に、PHTアクセス制御回路84は、決定97でDLBのビットがセットされていると決定したPHT53のセクタ67に書き込み信号を順次出す。プロセス98で、PHTアクセス制御回路84とマルチプレクサ82は、決定97

で示されたPHT53のそれぞれについて書き込み動作を繰り返す。各繰返しにおいて線R/Wを通して書き込み信号をセクタ67に与えると、選択されたPHT53にTSS90_jの対応するエントリ91の内容がロードされ、タスク_jを中断したタスク切替えのときに前に記憶した分岐パターン履歴に基づく予測コードを持つ。多数のPHT53をこのようにして書き換えるので、同じタスク内の異なるプログラムの種類（例えば、アプリケーションコードやライブラリ）の分岐命令は、タスク特有の分岐パターン履歴に基づく予測情報に基づいて予測することができる。プロセス98の後、流れは適当なタスク切替えルーチンに戻り、従来の方法でタスク切替えプロセスを終わる。

【0070】したがってこの発明の好ましい実施の形態の動作の結果、分岐パターン履歴に基づく予測情報は各タスクの専用の形式で保持され、他のタスク内の命令の分岐行動により予測情報が修正されるのを防ぐ。更にこの発明の好ましい実施の形態では、分岐パターン履歴に基づく予測情報の記憶と検索はタスク切替えのときに自動的に行われるので、プログラマがプログラム制御により行わせる必要はない。このようにこの発明により分岐予測性能が改善される。

【0071】この発明については種々の別の形態が考えられる。例えば、多数のPHTを用いることが好ましいが、この発明は、極端な場合は単一のグローバルPHTを用い、また逆の極端な場合は各BTBエントリ毎に1つのPHTを用いるなど、他のBTB構成で用いることができる。しかし単一のPHTを用いる場合は、タスクの開始のときに分岐パターン履歴に基づく予測情報を与える必要がある。このため、例えば新しいタスクへのタスク切替えのときにPHTの内容を記憶する。しかしそのタスクが分岐を予測するだけの十分な情報を持たない場合はPHTの内容に上書きしない。

【0072】この発明の他の実施の形態も考えられる。例えば、特にタスク状態セグメントを持たないアーキテクチャでは、タスク切替えのときにポインタを書き換えて、PHTと同等のもの、即ち分岐パターン履歴に基づく予測情報、が与えられるメモリ内の位置を指すようにしてよい。この例では、別のタスクはその疑似PHTとは別の位置を持ち、タスク切替えのときに単にポインタを書き換える。この方法は、オペレーティングシステムによる命令制御の下で処理するのが最もよい。

【0073】更に別の方法は、上に述べたように1つ以上のPHTをBTB内に設けて、機械状態レジスタ(MSR)への読取り及び書き込みと同様に、プログラム制御の下でロード及び記憶の操作に用いることである。この方法はオペレーティングシステムによる命令制御の下でも同様に実行できる。

【0074】更に、任意の上述の実施の形態及びその代替と組み合わせて、タスク切替えのときに分岐パターン

履歴に基づく予測情報の記憶やロードを選択的に可能にまた不可能にすることができる。例えば、MSRに可能／不可能ビットを設けて、その状態により予測情報の記憶及びロード動作が可能か不可能かを示す。この可能／不可能ビットのセットとリセットはプログラム制御の下で行う。

【0075】この発明について好ましい実施の形態を参照して説明したが、当業者はこの明細書と図面を参照することにより、この発明の特徴と利点を実現するこれらの実施の形態の修正や代替を考えることができる。このような修正や代替はこの発明の特許請求の範囲内にあるものである。

【0076】以上の説明に関して更に以下の項を開示する。

(1) 多重タスクモードで動作するマイクロプロセッサであって、第1及び第2のタスクに従って命令を実行する少なくとも1つの実行ユニットと、前記第1及び第2タスクのそれぞれに関連する部分を含み、また命令を記憶する部分を含む、メモリと、メモリにアドレスして前記実行ユニットが実行する命令コードを検索する取出しユニットであって、前記実行ユニットが実行した分岐命令の一連の結果を記憶する分岐履歴回路と、前記分岐履歴回路に結合し、前記分岐履歴回路からの分岐履歴フィールドに対応する予測情報を与える、パターン履歴回路と、取り出す命令のアドレスを選択する、アドレス指定回路と、を備える取出しユニットと、前記パターン履歴回路と前記メモリに結合し、前記第1タスクから前記第2タスクへのタスク切替えに応じて前記予測情報を修正する、回路と、を備える、マイクロプロセッサ。

【0077】(2) 前記メモリは前記第1及び第2タスクにそれぞれ関連する第1及び第2タスク制御構造を備え、前記パターン履歴回路は複数の指標付けされた予測コードエントリを持ち、また前記分岐履歴回路の分岐履歴フィールドに対応する予測コードエントリの1つの内容を出す出力を持ち、また前記修正回路は、パターン履歴回路からの予測コードエントリを前記メモリに送って前記第1タスク制御構造内に記憶し、また前記第1タスクから前記第2タスクへのタスク切替えに応じて、予測コードエントリを第2のタスク制御構造からパターン履歴回路に送る、回路、を備える、第1項記載のマイクロプロセッサ。

【0078】(3) 前記少なくとも1つの実行ユニットは前記メモリとの間でデータのロードと記憶を行うロード・記憶ユニットを備え、また前記通信回路は前記パターン履歴回路と前記ロード・記憶ユニットに結合するバスを備える、第2項記載のマイクロプロセッサ。

(4) 前記分岐履歴回路は、複数のエントリを有し、各エントリは関連する分岐命令の命令アドレスに対応するタグフィールドを持ち、またその関連する分岐命令の一連の前の分岐を記憶する分岐命令フィールドを持つ、

分岐ターゲットバッファ、を備える、第2項記載のマイクロプロセッサ。

(5) 前記分岐ターゲットバッファ内の複数のエントリは分岐ターゲットアドレスを記憶するターゲットフィールドをそれぞれ有し、また前記アドレス指定回路は、分岐を行う予測に対応する出力を前記選択論理が出すとこれに応じて、前記関連する分岐命令に対応するエントリの分岐ターゲットアドレスに対応するアドレスを選択する、第4項記載のマイクロプロセッサ。

【0079】(6) 前記パターン履歴回路は、前記分岐履歴回路に結合し、それぞれ複数の指標付けされた予測コードエントリを持ち、またそれぞれ前記分岐履歴回路からの分岐履歴フィールドに対応する予測コードエントリの1つの内容を出す出力を持つ、複数のパターン履歴表、を備え、前記実行ユニットは複数のプログラムの種類に従って命令を実行し、また前記取出しユニットは、プログラム種類標識を受けるように結合し、前記プログラム種類標識に対応する前記複数のパターン履歴表の1つの出力を前記アドレス指定回路に選択的に送る、選択論理、を更に有する、第2項記載のマイクロプロセッサ。

【0080】(7) 分岐命令のプログラム種類標識は前記分岐命令を含むプログラムに対応する特権レベル標識を備える、第6項記載のマイクロプロセッサ。

(8) 前記特権レベル標識は前記分岐命令に対応する1ビットのコードセグメントレジスタを備える、第7項記載のマイクロプロセッサ。

(9) 分岐命令のプログラム種類標識は、前記分岐命令を含むメモリの一部として少なくとも1ビットのページテーブルエントリを有する、第6項記載のマイクロプロセッサ。

【0081】(10) 前記タスク制御構造はそれぞれ、タスク切替えに応じて前記パターン履歴回路からの予測コードを前記タスク制御構造からロードするかどうかを示す状態を記憶する、動的ロード制御ビットと、前記パターン履歴回路からの予測コードを記憶するメモリ位置、を有する、第2項記載のマイクロプロセッサ。

【0082】(11) 前記パターン履歴回路は、前記分岐履歴回路に結合し、それぞれ複数の指標付けされた予測コードエントリを持ち、またそれぞれ前記分岐履歴回路からの分岐履歴フィールドに対応する予測コードエントリの1つの内容を出す出力を持つ、複数のパターン履歴表、を備え、前記実行ユニットは複数のプログラムの種類に従って命令を実行し、前記取出しユニットは、プログラム種類標識を受けるように結合し、前記プログラム種類標識に対応する前記複数のパターン履歴表の1つの出力を前記アドレス指定回路に選択的に送る、選択論理、を更に備え、また前記タスク制御構造はそれぞれ、タスク切替えに応じて前記複数のパターン履歴表の関連する1つの予測コードを前記タスク制御構造からロ

ードするかどうかを示す状態を記憶する、複数の動的ロード制御ビットと、前記複数のパターン履歴表の関連する1つからの予測コードをそれぞれ記憶する、複数のメモリ位置と、を備える、第2項記載のマイクロプロセッサ。

【0083】(12) 前記メモリは、前記マイクロプロセッサの外部にある主メモリと、前記マイクロプロセッサと共にオンチップであるキャッシュメモリと、を備え、前記タスク制御構造は主メモリに記憶され、また前記キャッシュメモリは前記タスク制御構造の写しを含む、第2項記載のマイクロプロセッサ。

(13) パイプライン化マルチタスクのマイクロプロセッサを操作する方法であって、パイプライン化マイクロプロセッサの取出し段階で第1のタスクの分岐命令を検出し、前記検出ステップに応じて、分岐履歴フィールドの少なくとも一部を検索し、前記分岐履歴フィールドの前記検索された部分に対応する記憶された予測情報から、分岐予測を生成し、前記第1タスクから第2のタスクへのタスク切替えに応じて、前記予測情報を修正する、パイプライン化マルチタスクマイクロプロセッサを操作する方法。

【0084】(14) 分岐予測を生成する前記ステップは、前記検出された分岐命令の分岐履歴フィールドの検索された部分に対応する、パターン履歴表内に記憶されている予測情報を検索することを含み、前記修正するステップは、前記パターン履歴表からの予測情報を前記第1タスクに関連するメモリの第1のタスク制御構造部分に記憶し、前記第2タスクに関連するメモリの第2のタスク制御構造部分から予測情報をロードする、第13項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

【0085】(15) タスク切替えを行う前記ステップは、メモリの前記第1タスク制御構造部分内の動的ロードビットを調べる、ことを更に含み、また前記記憶するステップは、メモリの前記第1タスク制御構造部分内の動的ロードビットが予測情報をメモリの前記第1タスク制御構造部分に記憶すべきことを示すと、これに応じて行う、第14項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

(16) タスク切替えを行う前記ステップは、メモリの前記第2タスク制御構造部分内の動的ロードビットを調べる、ことを更に含み、また前記ロードするステップは、メモリの前記第2タスク制御構造部分内の動的ロードビットが、予測情報がメモリの前記第2タスク制御構造部分にあることを示すと、これに応じて行う、第15項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

【0086】(17) 分岐予測を生成する前記ステップは、複数のパターン履歴表の中の選択された1つから検索された分岐履歴フィールドの部分に対応して行い、

タスク切替えを行う前記ステップは、メモリの前記第1及び第2タスク制御構造部分内の、それぞれ前記複数のパターン履歴表の1つに関連する複数の動的ロードビットを調べる、ことを更に含み、前記記憶するステップは、メモリの前記第1タスク制御構造部分内の1つ以上の動的ロードビットが、前記複数のパターン履歴表の関連するものについて予測情報をメモリの前記第1タスク制御構造部分に記憶すべきことを示すと、これに応じて行い、また前記ロードするステップは、メモリの前記第2タスク制御構造部分内の1つ以上の動的ロードビットが、前記複数のパターン履歴表の関連するものについて予測情報がメモリの前記第2タスク制御構造部分にあることを示すと、これに応じて行う、第14項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

【0087】(18) 前記分岐命令に対応するプログラムの種類を決定することを更に含み、また分岐予測を生成する前記ステップは、前記決定ステップで決定された前記プログラムの種類に従って選択された、複数のパターン履歴表の1つから検索された前記分岐履歴フィールドの部分に対応して行う、第14項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

(19) 前記決定するステップは、前記検出された分岐命令を含むプログラムに対応する特権レベル標識の状態を調べる、ことを含む、第18項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

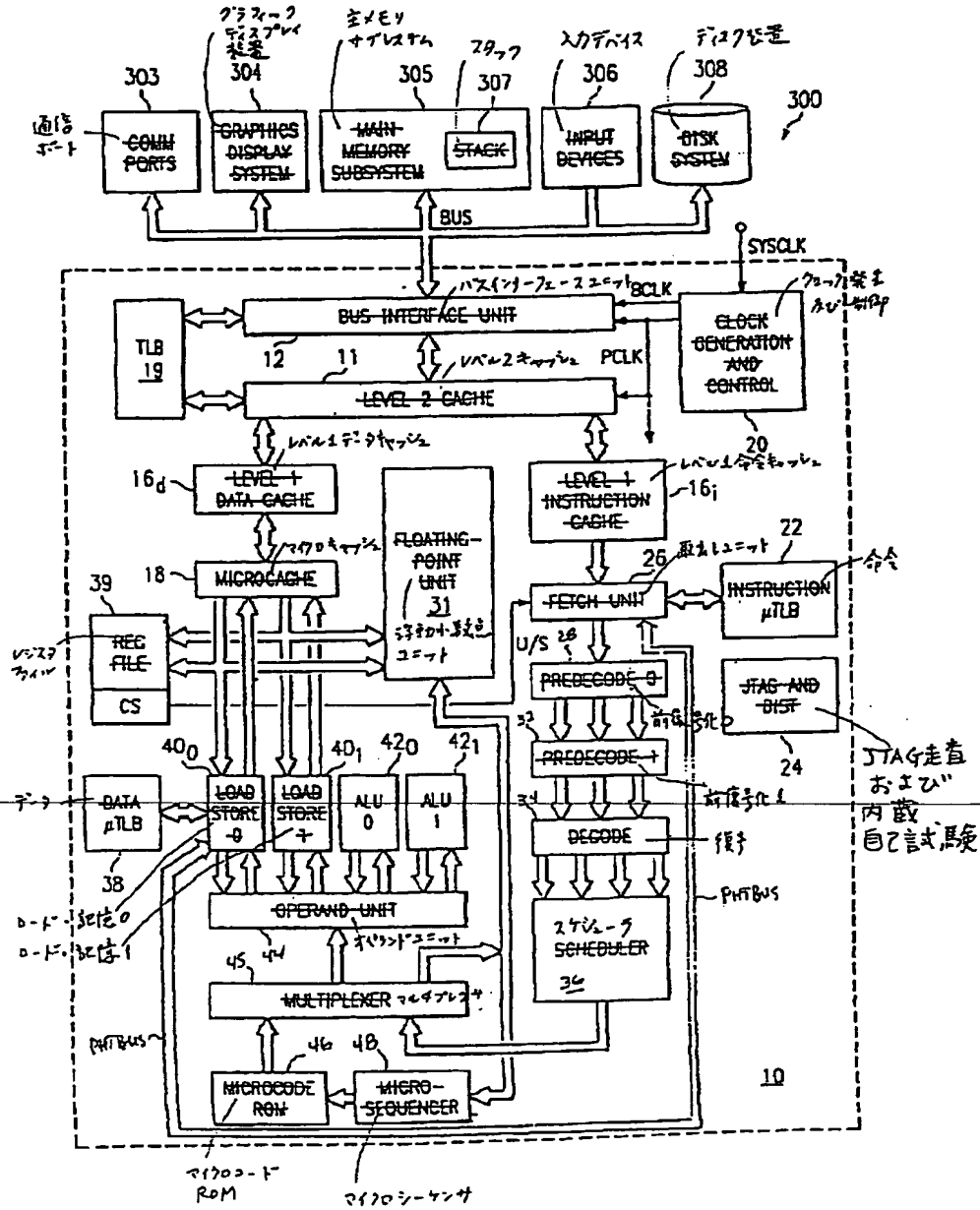
【0088】(20) 前記特権レベル標識は、前記検出された分岐命令に対応する1ビットのコードセグメントレジスタを備える、第19項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

(21) 前記決定するステップは、前記検出された分岐命令を含むメモリの一部について少なくとも1ビットのページテーブルエントリの状態を調べる、ことを含む、第18項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

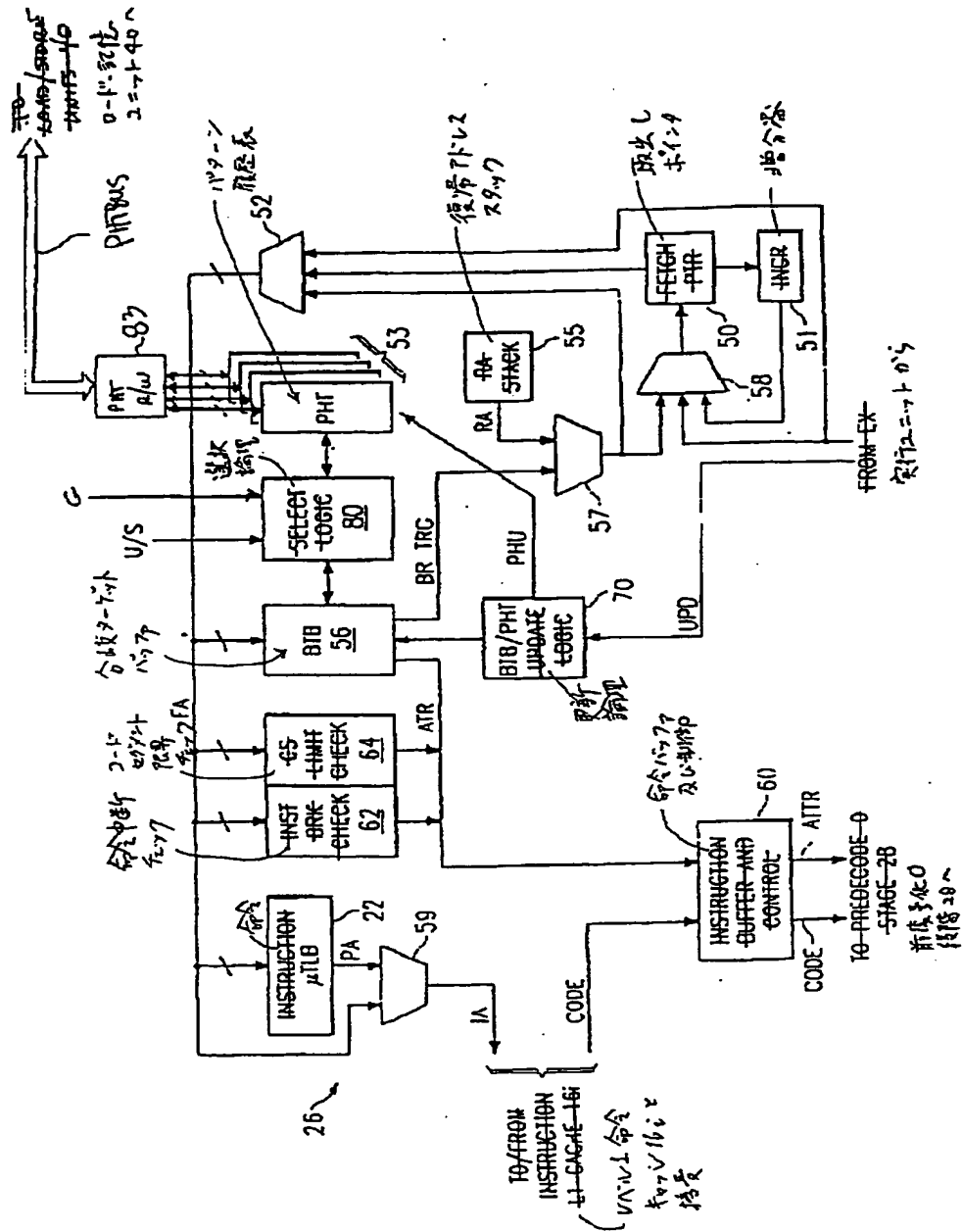
【0089】(22) 前記第1及び第2タスクの予測情報はメモリの第1及び第2部分にそれぞれ記憶され、前記生成するステップは、前記分岐履歴フィールドの検索された部分に対応するメモリの前記第1部分から予測情報を検索することを含み、前記修正するステップは、メモリの前記第2部分を指すようポインタを書き直し、前記第2タスク内の分岐命令を検出すると、前記生成するステップはメモリの前記第2部分から予測情報を検索するようにする、ことを含む、第13項記載のパイプライン化マルチタスクマイクロプロセッサを操作する方法。

【0090】(23) マイクロプロセッサとこれを含むシステムであって、分岐命令を含むプログラムの種類に応じて分岐予測を行う。取出しユニット(26)は分岐ターゲットバッファ(56)と、複数のパターン履歴

【図1】



【圖 2】



【図 4】

91 ₃	PHT 53,	+2A
91 ₂	PHT 53L	+2B
91 ₁	I970-2 PHT 53,	+2C
94	PHT AREA BASE .. N/U 018	+2D
	I/O ADDRESS 937 N/U IT	+2E
	N/U TASK LIST SELECTOR 0179	+2F
	N/U GS SELECTOR 0179	+2G
	N/U FS SELECTOR 0179	+2H
	N/U DS SELECTOR 0179	+2I
	N/U SS SELECTOR 0179	+2J
	N/U CS SELECTOR 0179	+2K
	N/U ES SELECTOR 0179	+2L
	EDI	
	ESI	
	EBP	
	ESP	
	EBX	
	EDX	
	ECX	
	EAX	
	EFLAGS	
	EIP	
	CR3 (PDBR)	
	N/U SS FOR CR2 OR SS	
	ESR FOR CR2 OR ESP	
	N/U SS FOR CPL1 OR SS	
	ESP FOR CPL1 OR ESP	
	N/U ES FOR CPL0 OR SS	
	ESP FOR CPL0 OR ESP	
	N/U BACK LINK TO FROM TSS	

31 115 116

TSS n-2

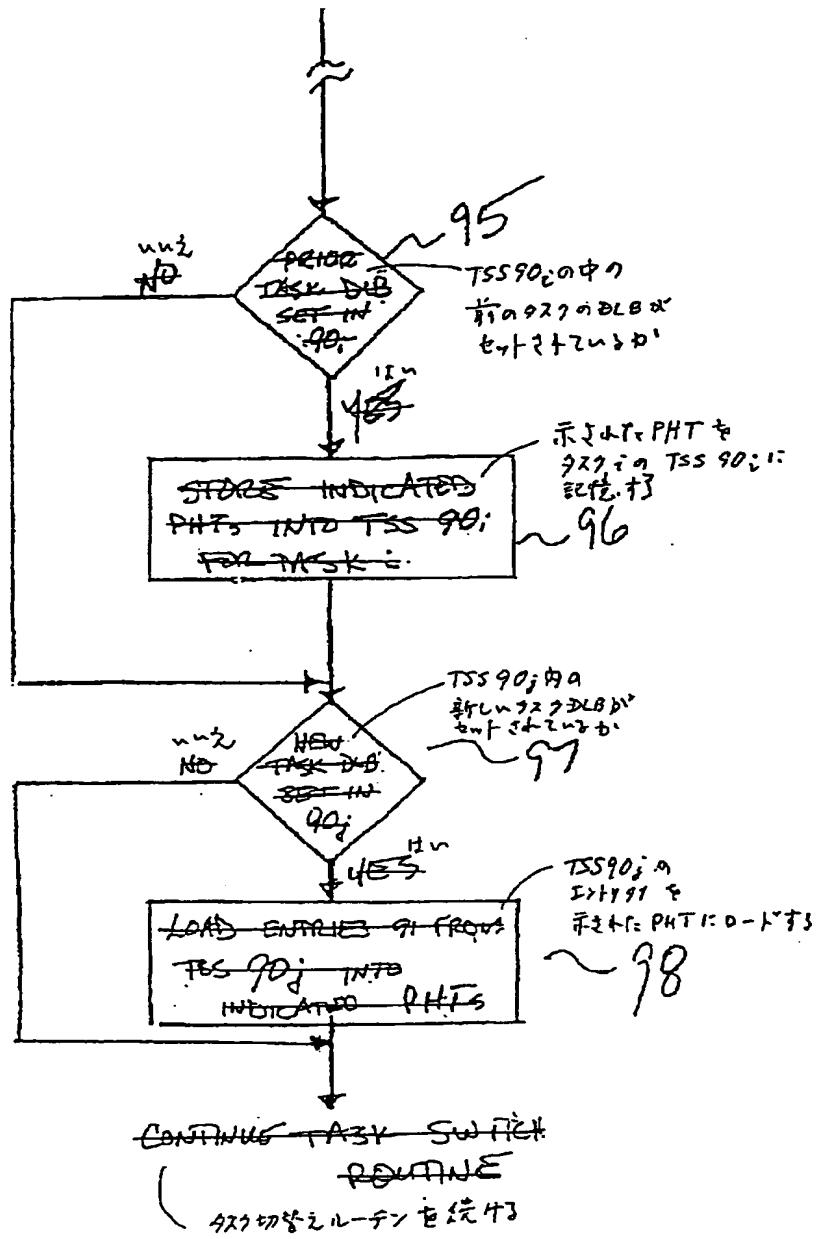
BASE

n-2

word
offset
from TSS
base
TSS n-2
is a
pointer

90

【図5】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.